

Association for Information Systems AIS Electronic Library (AISeL)

PACIS 2009 Proceedings

Pacific Asia Conference on Information Systems
(PACIS)

Summer 8-3-2009

VERTICAL DIFFERENTIATION IN THE MARKET FOR SECURITY SOFTWARE

Debabrata Dey

University of Washington - Seattle Campus, ddey@u.washington.edu

Guoying Zhang

Midwestern State University, grace.zhang@mwsu.edu

Follow this and additional works at: <http://aisel.aisnet.org/pacis2009>

Recommended Citation

Dey, Debabrata and Zhang, Guoying, "VERTICAL DIFFERENTIATION IN THE MARKET FOR SECURITY SOFTWARE" (2009). *PACIS 2009 Proceedings*. 27.
<http://aisel.aisnet.org/pacis2009/27>

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

VERTICAL DIFFERENTIATION IN THE MARKET FOR SECURITY SOFTWARE

Debabrata Dey

Professor of Information Systems and Evert McCabe Fellow
Michael G. Foster School of Business
University of Washington
Seattle, WA 98195, US
ddey@u.washington.edu

Guoying Zhang

Assistant Professor of Information Systems
Dillard College of Business
Midwestern State University
Wichita Falls, TX 76308, US
grace.zhang@mwsu.edu

Abstract

The market for security software has witnessed an unprecedented growth in recent years. A closer examination of this market reveals that, unlike a traditional software market, the use of vertical differentiation strategy is quite limited in this market. In this paper, we develop a quantitative model to explore the possible reason. Our model identifies a negative network externality effect as the primary reason for this divergence. Using our model, we show that, in this market, the vertical differentiation strategy would never be employed by a monopolist. We then extend our analysis to a duopoly competition and find that, although vertical differentiation may be adopted if the cost of development is sufficiently high, due to the presence of the negative network externality, the feasible region for differentiation is much more restricted when compared to a traditional market.

Keywords: Security software, network externality, negative network effect, vertical differentiation.

1. INTRODUCTION

The industry of security software continues to grow quite rapidly in response to the increasing demand for the protection of a growing base of information technology (IT) infrastructure. The worldwide security software revenue is expected to increase from nearly \$8.3 billion in 2006 to more than \$13.5 billion in 2011, at a compound annual rate of 10.4% (Latimer-Livingston and Contu 2007). The security software market in Asia (excluding Japan) alone has demonstrated about 23% growth in 2005 and is expected to reach US\$1.7 billion by 2010 (Low and Chung 2006). Security software market thus has been regarded as one of a few prominent software markets with double-digit growth rate (McCormack 2006). Understanding the nature of this market, along with the appropriate product strategy, is of importance to vendors as well as consumers.

In general, security software can be classified into several categories, such as antivirus software, encryption software, firewall and intrusion detection/protection systems (IDS or IPS), and spyware remover. Antivirus software is perhaps the most well-known type of security software, primarily used to identify and remove viruses, but can often provide protection against other malicious invasions, such as worms, phishing attacks, and Trojans. Encryption software is used to encrypt computer data using an encryption algorithm. With the proliferation of broadband network, security of online data exchange becomes a key determinant to the success of E-commerce vendors. By deploying appropriate encryption software on its Web infrastructures, an E-commerce vendor can relieve customers from the concerns on privacy and security and thus build up a reputation of trust. Firewalls are used in local area networks to inspect the traffic going to or coming from an outside network and make decisions about whether a transmission should be allowed. Nowadays, Microsoft bundles a firewall with its Windows operating systems that can run even on a standalone personal computer. A spyware remover is a tool to detect and remove spyware—a piece of software that gets installed on a computer, usually through the Internet, without the user's permission. Although a spyware does not paralyze the computer or make modifications to the data, it can monitor the user's behaviour, collect various types of personal information, and pass the information to the party initiating the installation of the spyware.

In a traditional software market, users usually enjoy a higher network utility derived from a larger market share, which is often referred to as the *positive network externality* (Katz and Shapiro 1985, 1986). This positive network externality primarily arises from users' need for compatibility, which allows users to share files and information, edit and critique documents created by others, and, most importantly, work in a collaborative setting. It is well-known that positive network externality can lead to a near-monopoly market condition: if a vendor's market share is large enough to exceed a critical mass, other competitors will lose opportunities to enter the market. Studies on the markets for operating systems and application software (such as Spreadsheet and Word processor) empirically validate this near-monopolistic structure (Brynjolfsson and Kemerer 1996, Liebowitz and Margolis 1999). However, the security software market is markedly different. It is characterized by many vendors, with no single dominant player. For example, the market of anti-virus software has several major players including Symantec, McAfee, Trend Micro, Computer Associates, and Panda Software, besides dozens of other smaller companies. Fosfuri and Giarratana (2004) found that, between 1989 and 1998, 270 vendors entered this market, with a very high percentage not surviving beyond two years (Giarratana 2004).

In practice, when head-to-head market competition is fierce, vendors often resort to *vertical product differentiation* (Gabszewicz and Thisse 1980, Moorthy 1984, 1988). Even in the near-monopolistic traditional software market, examples abound where the vendor sells several different versions of the same product (Bhargava and Choudhary 2001, Hui et al. 2007-08, Raghunathan 2000). For example, Microsoft packages basically the same Windows operating system and Office application suite differently for home and professional users. Similarly, Oracle offers Oracle 10g and Oracle 10g Express versions to target different market segments. On the other hand, in the security software market, vendors do not always offer a degraded or "express" version simultaneously with the full version of the product. Most of the time, the seemingly different versions are essentially different

bundles of several component products. For example, Symantec offers three different bundles: Norton AntiVirus, Norton 360, and Norton 360 Premier—Norton 360 bundles anti-spyware with Norton Antivirus, and Norton 360 Premier bundles spyware and phishing protection with Norton Antivirus. Other variations of security software are often different versions based on a timeline—the yearly upgraded version is not really a simultaneous offer for a different market segment; rather, it represents continuous product improvement. This observation naturally leads to our research question—despite its highly competitive nature, why the use of vertical differentiation strategy is so limited in the security software market?

The objective of this research is to develop a quantitative model to address this research question. We first note that, unlike traditional software markets, positive network externality is not observed in the security software market. In fact, from the perspective of a user, security software is simply used to prevent security exploitations, and there is hardly any benefit from the compatibility of user data. Instead, our analysis finds a negative network externality effect in this market. When a user adopts a security software, there are two benefits: (i) a direct benefit—representing the mitigation effect on direct security attacks by hackers, and (ii) an indirect benefit—arising from the prevention of indirect attack or infection from other users in the network (Ogut et al. 2005). In an indirect attack, a system is not a direct target, but could become an eventual target from the security exploitation of another system. Typical examples of indirect attacks include the prevalence of Internet worms (Braverman 2005) and the wide presence of BOT net agents (Sancho 2005), which could launch large-scale attack with the ability to convert ordinary nodes into malicious agents. The user's indirect benefit eventually leads to a negative network effect—the larger the market coverage of security software, the less is the indirect benefit because the indirect threats are already mitigated, and the chance of getting infected from others reduces. Such indirect effects have also been recognized by Anderson (2001) as the “tragedy of commons,” by Png et al. (2006) as the “the reason of users’ inertia of taking security precautions,” and by August and Tunca (2006) as an important factor in changing the users’ incentive to apply security patches. Incorporation of this diminishing indirect benefit into our model leads to an increasingly less network valuation by users from a larger market coverage. We find that this negative network effect helps to explain the limited nature of vertical differentiation in this market.

We analyze both monopoly and duopoly markets. It is shown that vertical differentiation is not an attractive strategy to a monopolist, but it may be adopted in a duopoly if the development cost is sufficiently high. However, the negative network effect significantly shrinks the region of vertical differentiation in a duopoly. This study highlights the unique nature of the security software market and provides managerial insights for vendors on market competition and product development strategies.

The rest of the paper proceeds as follows. Section 2 develops the user model. Section 3 and 4 evaluate the strategy of vertical differentiation under monopoly and duopoly settings, respectively. We conclude in Section 5 and offer future research directions.

2. THE USER MODEL

For the sake of exposition, we first develop the user model under the assumption of no vertical differentiation and then extend it to the case of vertical differentiation.

2.1 No Vertical Differentiation

Consumers (users) of security software are assumed to be heterogeneous because the amount of benefit from thwarting an attack would vary from user to user. In order to capture this, consumers are indexed by a parameter u that indicates their relative expected benefit if an attack is thwarted; we assume that u is uniformly distributed over the interval $[0, 1]$. The absolute expected benefit to user u from thwarting an attack can then be expressed as Lu , where L is a constant. Lu can also be viewed as a proxy for the potential loss to user u from an attack (Gordon and Loeb 2002). As mentioned in Section 1, there are two types of benefits derived from adopting a security software—direct and indirect. First, consider the direct benefit. Assume that hackers could launch successful attacks on an unprotected system at an average rate of λ_D . Therefore, by adopting security software, user u has a

The total benefit (per unit time) to user u from adopting the software, in a market with coverage x , can then be written as:

where $g = \lambda/\lambda_D$. Clearly, the parameter g is a proxy for the negative network externality effect—the higher the g , the larger is the potential indirect benefit and, hence, the more significant is the negative network effect. Writing the above expression in this form provides us with the flexibility to easily capture various levels of the relative indirect utility, which can be attributed to software characteristics as well as the network connectivity. For example, anti-virus and anti-spyware software have a higher indirect effect and hence a higher g , whereas an encryption software might have a lower g . A well-connected network is likely to have a higher value of g , when compared to a sparser network.

$$\lambda_p Lu(1+g(1-x))-P=0.$$
$$p = (1 + g(1 - x))(1 - x). \quad (1)$$

The diagram shows a horizontal line representing the range of possible values for u . The line starts at 0 and ends at 1. A vertical arrow points down from the text "Indifferent Consumer" to a point on the line labeled u . The line is divided into two regions by this point. The region to the left of u is labeled "Do Not Adopt" and "Fraction: $1 - x = u$ ". The region to the right of u is shaded gray and labeled "Adopt" and "Fraction: $x = 1 - u$ ".

Figure 1: Consumers Choose to Adopt (or Not Adopt) Based on Their Relative Benefits

2.2 Vertical Differentiation

We now incorporate vertical differentiation by extending the user model. Consider a market where, at the same time, two security software products are offered with similar functionalities, but different quality. The two products are characterized by a quality parameter, which can also be viewed as the effectiveness of the security software in providing the protection it is supposed to. We assume that the superior product has a quality of q_h , whereas the inferior version has a quality q_l , $0 < q_l \leq q_h \leq 1$. We model the normalized development cost of a product with quality level of q as cq^2 , where c is a constant.

The utility of a user who adopts one of the versions of the software changes in two ways: (i) the direct utility needs to be discounted by the quality parameter $q \in \{q_l, q_h\}$, and (ii) the indirect utility also needs to be modified because now the effective coverage of each version is discounted by $q \in \{q_l, q_h\}$. The overall market characterized by $u \sim \text{Uniform}(0, 1)$ can be segmented into three parts now by points u_h and u_l , where $0 \leq u_l \leq u_h \leq 1$. This is represented in Figure 2. The users in $(u_h, 1]$ choose the superior version at price p_h , the users in (u_l, u_h) choose the inferior version at price p_l , and the users in $[0, u_l)$ opt not to adopt either version. The respective market sizes for the superior and the inferior versions are: $x_h = 1 - u_h$ and $x_l = 1 - u_l$. Of course, the marginal users, u_h and u_l , must satisfy the following incentive compatibility and individual participation conditions:

$$\begin{aligned} q_l (1 + g(1 - q_h x_h - q_l x_l)) u_l - p_l &= 0, \text{ and} \\ q_h (1 + g(1 - q_h x_h - q_l x_l)) u_h - p_h &= q_l (1 + g(1 - q_h x_h - q_l x_l)) u_h - p_l. \end{aligned}$$

Substituting $u_h = 1 - x_h$ and $u_l = 1 - x_h - x_l$ into the above conditions and solving for the prices, we get:

$$p_l = q_l (1 - x_h - x_l) (1 + g(1 - q_h x_h - q_l x_l)), \text{ and} \quad (2)$$

$$p_h = (q_h (1 - x_h) - q_l x_l) (1 + g(1 - q_h x_h - q_l x_l)). \quad (3)$$

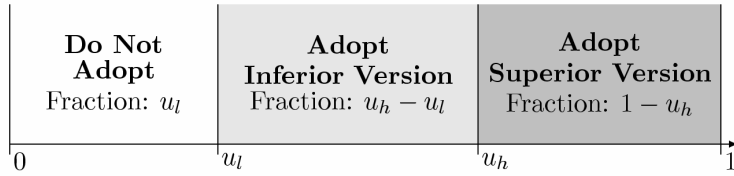


Figure 2: Segmentation of the Consumer Market

3. VERTICAL DIFFERENTIATION IN A MONOPOLY MARKET

Consider a monopolist who wants to offer two versions of a security software product characterized by quality parameter $q \in \{q_l, q_h\}$. As before, we model the development cost of quality q as cq^2 . However, since the vendor is offering two versions of basically the same product, we assume that the vendor only incurs the development cost for the superior product and does not incur any additional cost for the inferior version. This makes sense since the additional production and updating costs are negligible. The additional development cost is also minimal since the vendor can simply turn off a few of the advanced features to provide the inferior version (Raghunathan 2000).

Using (2) and (3), the total profit for the monopolist can be calculated as:

$$R_{\text{mon}} = p_l x_l + p_h x_h - cq_h^2 = (q_l x_l (1 - 2x_h - x_l) + q_h x_h (1 - x_h)) (1 + g(1 - q_h x_h - q_l x_l)) - cq_h^2.$$

The vendor's profit maximization problem can then be written as:

$$\begin{aligned} \text{Max}_{x_l, x_h, q_l, q_h} R_{\text{mon}} &= (q_l x_l (1 - 2x_h - x_l) + q_h x_h (1 - x_h)) (1 + g(1 - q_h x_h - q_l x_l)) - cq_h^2 \\ \text{s.t. } &0 < q_l \leq q_h \leq 1, \quad 0 \leq x_l + x_h \leq 1. \end{aligned} \quad (4)$$

Proposition 1: *In a security software market, a monopolist would not employ a vertical differentiation strategy.*

Proof: To solve (4), we consider the following two first-order conditions: $\frac{\partial R_{\text{mon}}}{\partial x_h} = 0$ and $\frac{\partial R_{\text{mon}}}{\partial x_l} = 0$.

Combining, we get:

$$q_l \frac{\partial R_{\text{mon}}}{\partial x_h} - q_h \frac{\partial R_{\text{mon}}}{\partial x_l} = 2q_l(q_h - q_l)x_l(1 + g(1 - q_h x_h - q_l x_l)) = 0.$$

This has four distinct roots: $q_l=0$, $q_h=q_l$, $x_l=0$, or $(1+g(1-q_h x_h - q_l x_l))=0$. Since $q_l > 0$, the first root can be discarded. The last one implies that $q_h x_h + q_l x_l = 1 + 1/g > 1$, which is impossible since $x_l + x_h \leq 1$ and $q_l, q_h \leq 1$; thus, the last root must also be discarded. Therefore, either $q_h=q_l$ —there is no vertical differentiation—or $x_l=0$ —there is no market for the inferior version. In either case, the monopolist does not employ the vertical differentiation strategy. ■

Clearly, this result is quite different from theoretical results and practical observations in a monopoly software market, where product versioning with different prices is a common strategy to capture the marginal users. The lack of positive externality effect in the security software market makes such a strategy sub-optimal.

4. VERTICAL DIFFERENTIATION IN A DUOPOLY MARKET

We now examine whether the strategy of vertical differentiation would be adopted in a duopoly market. We use a traditional setup for vendors' differentiation choices: each vendor selects a quality level $q \in \{q_l, q_h\}$ and a price $p \in \{p_l, p_h\}$ to compete in the market, and users make rational choices of the products followed by the realization of payoffs. The prices charged must abide by the conditions in (2) and (3). The high-quality provider then solves the following optimization problem:

$$\begin{aligned} \text{Max}_{x_h, q_h} \quad & R_h = p_h x_h - c q_h^2 = x_h (q_h (1 - x_h) - q_l x_l) (1 + g(1 - q_h x_h - q_l x_l)) - c q_h^2 \\ \text{s.t.} \quad & 0 < q_h \leq 1, \quad 0 \leq x_l + x_h \leq 1, \end{aligned} \quad (5)$$

while the low-quality provider solves:

$$\begin{aligned} \text{Max}_{x_l, q_l} \quad & R_l = p_l x_l - c q_l^2 = x_l q_l (1 - x_h - x_l) (1 + g(1 - q_h x_h - q_l x_l)) - c q_l^2 \\ \text{s.t.} \quad & 0 < q_l \leq q_h, \quad 0 \leq x_l + x_h \leq 1. \end{aligned} \quad (6)$$

		Constraint $q_h \leq 1$	
		Binding	Slack
Constraint $q_l \leq q_h$	Binding	Region I $q_l = 1$ $q_h = 1$	Region IV $q_l = q_h$ $q_h < 1$
	Slack	Region II $q_l < q_h$ $q_h = 1$	Region III $q_l < q_h$ $q_h < 1$

Figure 3: Four Feasible Regions for Equilibrium Outcome

Because of the development cost asymmetry in this case, differentiation is a possible strategy, especially when the development cost is high. In order to analyze this case in a more rigorous fashion,

we decompose the feasible region of the equilibrium outcome into four regions, as shown in Figure 3. It is clear that the two vendors would use vertical differentiation in Regions II and III, whereas they would not differentiate the products in Regions I and IV.

Lemma 1: *An equilibrium outcome cannot be in Region IV.*

Proof: We will prove this by contradiction. Let an equilibrium solution in Region IV be $q_l = q_h = q < 1$. In this region, $q_h < 1$, so the high-quality provider solves (5) without the quality constraint, and the solution must satisfy the following first-order condition:

$$\frac{\partial R_h}{\partial q_h} = x_h(1 - x_h)(1 + g(1 - q_h x_h - q_l x_l)) - g x_h^2(q_h - q_h x_h - q_l x_l) - 2c q_h = 0.$$

Since $q_l = q_h = q$, the two vendors should have equal market share; let $x_l = x_h = x$. The above condition then reduces to:

$$g q x^2(1 - 2x) + 2c q = x(1 - x)(1 + g(1 - 2q x)). \quad (7)$$

We now turn our attention to how the revenue of the low-quality provider changes with the quality of her own product; from (6):

$$\frac{\partial R_l}{\partial q_l} = x_l(1 - x_h - x_l)(1 + g(1 - q_h x_h - q_l x_l)) - g x_l^2 q_l(1 - x_h - x_l) - 2c q_l.$$

Once again, setting $q_l = q_h = q$ and $x_l = x_h = x$, and substituting (7), we get:

$$\left. \frac{\partial R_l}{\partial q_l} \right|_{(q,x)} = x(1 - 2x)(1 + g(1 - 2q x)) - x(1 - x)(1 + g(1 - 2q x)) = -x^2(1 + g(1 - 2q x)) < 0. \quad (8)$$

Since $q > 0$, (8) simply means that the low-quality provider can increase her profit by simply decreasing q_l from q , thereby moving into Region III. Of course, since this new solution abides by the constraint $q_l \leq q_h$, it is a valid move by the low-quality provider. Furthermore, such a move by the low-quality provider is beneficial to the high-quality provider as well; this is because:

$$\left. \frac{\partial R_h}{\partial q_l} \right|_{(q,x)} = -x^2(1 + g(1 - q)) < 0.$$

Clearly then, the equilibrium outcome could not have been in Region IV. ■

Our analysis shows that the equilibrium outcome can occur in any of the other three regions. The actual outcome depends on c , the development cost parameter. This dependence can be understood intuitively; see Figure 4. First, when c is low, i.e., the cost associated with developing a high-quality product is still low, both the vendors choose the highest level of quality— $q_l = q_h = 1$; there is no vertical differentiation, and the equilibrium is observed in Region I. However, as c increases beyond a threshold (γ_1), the high cost of quality forces one of the vendors to cut down on the development cost by lowering the quality ($q_l < 1$), while the other vendor maintains the high quality level ($q_h = 1$)—the equilibrium shifts to Region II, and product differentiation is observed. As c increases further, beyond a second threshold (γ_2), the high-quality vendor is also forced to reduce the quality level ($q_h < 1$), but, as shown in Lemma 1, she always maintains a quality level higher than that of the other vendor ($q_h > q_l$); the equilibrium is observed in Region III.

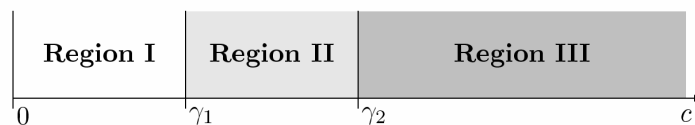


Figure 4: *Equilibrium Outcome Region Changes with c*

It may first appear from Figure 4 that the equilibrium outcome depends only on c , and not on the negative network externality parameter, g . However, g has an important role to play in determining the equilibrium outcome and, hence, the product differentiation strategy. In order to understand the role played by g , we need to determine the two thresholds, γ_1 and γ_2 .

Lemma 2: *The boundary between Regions I and II is characterized by the following threshold on c :*

$$\gamma_1 = \frac{9 + 26g + 52g^2 + 8g^3 + (4g^2 - 8g - 3)\sqrt{4g^2 + 4g + 9}}{1024g^2} \quad (9)$$

Proof: In both the regions, $q_h=1$; the low-quality vendor's optimization problem can, therefore, be simplified to:

$$\text{Max}_{x_l, q_l} R_l = x_l q_l (1 - x_h - x_l) (1 + g(1 - x_h - q_l x_l)) - c q_l^2; \quad 0 < q_l, x_l + x_h \leq 1.$$

The following first-order condition must be satisfied by the solution of the unconstrained problem:

$$\frac{\partial R_l}{\partial q_l} = x_l (1 - x_h - x_l) (1 + g(1 - x_h - q_l x_l)) - g x_l^2 q_l (1 - x_h - x_l) - 2c q_l = 0.$$

Solving this, we get:

$$q_l = \frac{(1 + g(1 - x_h))(1 - x_h - x_l) x_l}{2(c + g x_l^2 (1 - x_h - x_l))}.$$

Now, at the boundary of Regions I and II, $q_l=1$. This would be the case if:

$$c = \frac{x_l (1 - x_h - x_l) (1 + g(1 - x_h - 2x_l))}{2} = \gamma_1. \quad (10)$$

Now, if, $q_l=q_h=1$, then the situation reduces to a duopoly case with both offering a product at $q=1$. A more general result for the oligopoly case is provided by Dey and Zhang (2009). Setting $n=2$ in that result, we get:

$$x_h = x_l = x = \frac{3(2g + 1) - \sqrt{4g + 4g^2 + 9}}{16g},$$

which can be substituted into (10) to obtain (9). ■

Conjecture 1: *The boundary between Regions II and III is characterized by the following threshold on c :*

$$\gamma_2 \approx \frac{9.3 + 5.35g}{1280(1 + g)^3} \left(16 + 33g + 23.5g^2 + 4g^3 + (0.3 + 2g)\sqrt{14 + 28g + 26g^2 + 14g^3 + 4g^4} \right). \quad (11)$$

Rationale: In Region III, all the four first-order conditions must be satisfied:

$$\frac{\partial R_h}{\partial x_h} = \frac{\partial R_h}{\partial q_h} = \frac{\partial R_l}{\partial x_l} = \frac{\partial R_l}{\partial q_l} = 0.$$

In addition, at the boundary of Regions II and III, $q_h = 1$. Substituting this into the above four conditions, we get four equations and four unknowns (x_h, x_l, q_l and c). From the first three equations, we can find that:

$$q_l = \frac{1 + g - x_h - 2gx_h + gx_h^2 - 2x_l - 2gx_l + 2gx_h x_l}{gx_l (2 - 2x_h - 3x_l)},$$

$$x_l = \frac{4c - x_h - 4cx_h - gx_h + x_h^2 + 4gx_h^2 - 5gx_h^3 + 2gx_h^4}{6c - x_h - gx_h - x_h^2 + 3gx_h^2 - 2gx_h^3}, \text{ and}$$

$$c = \frac{x_h}{4} \left(1 + 2gx_h - 4gx_h^2 + (1 - 2x_h) \sqrt{4g^2x_h^2 + 1} \right). \quad (12)$$

All these can be substituted sequentially into the last equation, and an equation involving only x_h can be obtained. This equation is highly non-linear and could not be solved analytically. However, we were able to solve it numerically. A very close approximation of this solution is given by:

$$x_h \approx 0.267647 + \frac{0.197238}{1+g}.$$

This is substituted into (12) to obtain (11). ■

The above analysis leads to our overall result:

Proposition 2: *In a security software market, two vendors would use a vertical differentiation strategy, if and only if:*

$$c > \frac{9 + 26g + 52g^2 + 8g^3 + (4g^2 - 8g - 3)\sqrt{4g^2 + 4g + 9}}{1024g^2}.$$

Proof: Vertical differentiation will be observed as long as the equilibrium outcome is not in Region I. The above condition ensures that this is indeed the case (see Lemma 2). ■

The result in Proposition 2 can be better visualized in Figure 5, where the (g, c) space is partitioned into two regions by γ_1 . It is clear from Figure 4 that product differentiation decision in a duopoly depends both on the development cost and the negative externality. The threshold separating the two decision choices (γ_1) is an increasing function of g . This implies that, although product differentiation is a valid strategy in the security software market, its feasible region shrinks significantly as g increases. Therefore, product differentiation is likely to be less predominant in this market when compared to a traditional market ($g=0$). This perhaps explains why the quality levels of competing products are so close to one another in this market.

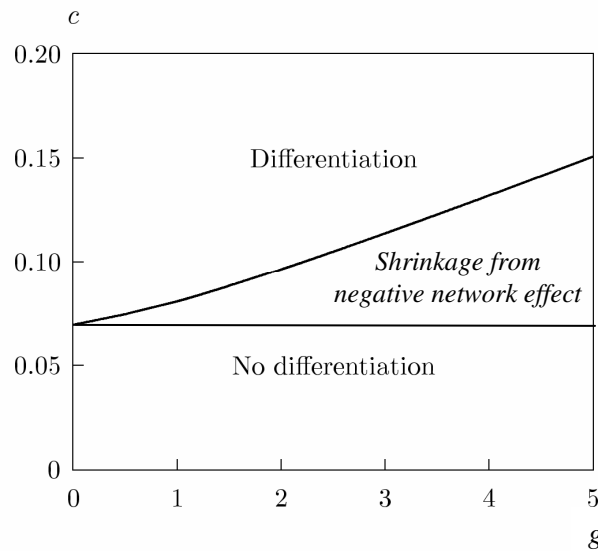


Figure 5: Product Differentiation Strategy as a Function of c and g

5 CONCLUSION AND FUTURE DIRECTIONS

A security software is a tool employed by individuals and organizations alike to prevent security exploitations of computerized systems. Over the last decade, the market for this type of software has seen a tremendous growth, both from the supply as well as the demand side. Unlike the traditional software market, where the supply side is dominated by only a few providers, the market for security software is highly competitive with several major players.

We find that the positive network externality enjoyed by traditional software is much weaker for security software; this is because the compatibility issue of application data across users is not a big concern. On the contrary, we find that there is a negative network externality effect for security software. This negative effect is derived from the fact that, as the market coverage grows, the chance of an indirect attack from an unprotected computer decreases.

Incorporating the negative network externality, we examine the strategy of vertical product differentiation in this competitive market. We find that this strategy would never be adopted in a monopoly market. Although, it may be adopted in a duopoly market if the cost of quality is sufficiently high, the feasible region of differentiation shrinks because of the negative network effect. Overall, our results capture the unique structure of the security software market, and provide insights for security software vendors on market competition and strategies.

Our research is the first of its kind in trying to explain the market behaviour of security software providers from an analytical angle. There are several directions in which our results can be extended. First, in traditional software market, nonlinear pricing through volume licensing is quite common. One could investigate how nonlinear pricing can help security software vendors leverage the network effect and how that will impact the result of vertical differentiation. Another promising direction is to study vertical differentiation when there are more than two vendors in the market. Finally, vendors from traditional software markets are entering the security software market, and Internet service providers are also offering security software to their subscribers. These new competition patterns would be another interesting area of study. We are examining some of these issues in our ongoing efforts to better understand this market.

References

- Anderson, J. 2001. Why Information Security is Hard—an Economic Perspective. *Proceedings of the Seventeenth Computer Security Applications Conference* 358-365.
- August, T., T.I. Tunca. 2006. Network Software Security and User Incentives. *Management Science* 52(11) 1703-1720.
- Bhargava, H., V. Choudhary. 2001. Information Goods and Vertical Differentiation. *Journal of Management Information Systems* 18(2) 89-106.
- Braverman, M. 2005. Win32/Blaster: A Case Study From Microsoft's Perspective. *Virus Bulletin Conference* 200-205. Available at: <http://download.microsoft.com>.
- Brynjolfsson, E., C. Kemerer. 1996. Network Externalities in Microcomputer Software: An Econometric Analysis of the Spreadsheet Market. *Management Science* 42(12) 1627-1647.
- Dey, D., G. Zhang. 2009. Competition in the Security Software Market. Working Paper, University of Washington, Seattle, WA.
- Fosfuri, A., M.S. Giarratana. 2004. Product Strategies and Start-ups' Survival in Turbulent Industries: Evidence from the Security Software Industry. Working Paper, Universidad Carlos III de Madrid.
- Gabszewicz, J.J., J.-F. Thisse. 1980. Entry (and Exit) in a Differentiated Industry. *Journal of Economic Theory* 22(2) 327-338.

Giarratana, S.M. 2004. The Birth of a New Industry: Entry by Start-ups and the Drivers of Vendor Growth: The Case of Encryption Software. *Research Policy* 33(5) 787-806.

Gordon, L.A., M.P. Loeb. 2002. The Economics of Information Security Investment. *ACM Transactions on Information and System Security* 5(4) 438-457.

Hui, W., B. Yoo, K.Y. Tam. 2007-08. The Optimal Number of Versions: Why Does Goldilocks Pricing Work for Information Goods? *Journal of Management Information Systems* 24(3) 167-191.

Katz, M.L., C. Shapiro. 1985. Network Externalities, Competition and Compatibility. *The American Economic Review* 75(3) 424-440.

Katz, M.L., C. Shapiro. 1986. Technology Adoption in the Presence of Network Externalities. *Journal of Political Economy* 94(4) 822-841.

Latimer-Livingston, N.S., R. Contu. 2007. Forecast: Security Software Worldwide, 2006-2011, Update. *Gartner Report*. Available at: http://www.gartner.com/DisplayDocument?ref=g_search&id=510567&subref=sim.

Liebowitz, S.J., E.S. Margolis. 1999. Cause and Consequences of Market Leadership in Application Software. Conference of Competition and Innovation in the Personal Computer Industry, Carlsbad, California. April.

Low, W., D. Chung. 2006. Asia/Pacific (Excluding Japan) Security Software 2006—2010 Forecast and Analysis. *IDC Report*. Available at: <http://www.idc.com/getdoc.jsp?containerId=AP322306N>

McCormack, K. 2006. Enterprise Software's Growth Pocket. (An Interview of S&P's Zaineb Bokhari by Karyn McCormack). *BusinessWeek*, May 24. Available at: http://www.businessweek.com/investor/content/may2006/pi20060524_164054.htm.

Moorthy, K.S. 1984. Market Segmentation, Self-selection, and Product Line Design *Marketing Science* 3(4) 288-307.

Moorthy, K.S. 1988. Product and Price Competition in a Duopoly. *Marketing Science* 7(2) 141-168.

Ogut, H., N. Menon, S. Raghunathan. 2005. Cyber Insurance and IT Security Investment: Impact of Interdependent Risk. Presented at the 4th Workshop on the Economics of Information Security, Cambridge, MA, June 2-3.

Png, I.P.L., C.Q. Tang, Q. Wang. 2006. Hackers, Users, Information Security. Presented at the 5th Workshop on the Economics of Information Security, Cambridge, UK, June 26-28.

Raghunathan, S. 2000. Software Editions: An Application of Segmentation Theory to the Packaged Software Market. *Journal of Management Information Systems* 17(1) 87-113.

Sancho, D. 2005. The Future of BOT Worms. *Trend Micro White Paper*. Available at: http://us.trendmicro.com/imperia/md/content/us/pdf/threats/securitylibrary/future_of_bots_final.pdf.